

REMARKS

This Amendment and Response is intended to be fully responsive to the Office Action dated April 8, 2004. In that Office Action, Figures 1 and 8 were rejected for lacking a "Prior Art" legend; claims 10 and 23 were objected to because of informalities; claim 17 was rejected under 35 U.S.C. §112; and claims 1-23 were rejected under 35 U.S.C. § 103(a) as being unpatentable over "C++: Effective Object-Oriented Software Construction" by Dattatri in view of "Compiler Transformations for High-Performance Computing" by Bacon.

In this Amendment and Response, claims 10, 17, and 23 have been amended. No claims have been added, and no claims have been canceled. Reexamination and reconsideration of the above rejections and objections are respectfully requested in light of the above amendments and these remarks.

Drawings

FIGs. 1 and 8 were objected to as the Examiner alleges that these Figures should be "designated by a legend such as --Prior Art-- because only that which is old is illustrated." These Figures have not been amended herein because Applicant respectfully asserts that these figures do not represent "prior art." Rather, they represent part of the implementation of the invention. More specifically, FIG. 1 depicts an exemplary class hierarchy with methods that return constants in an embodiment of the present invention. FIG. 8 depicts a computer system suitable for practicing the invention. Both figures are provided to show an implementation of the invention, and provide an enabling disclosure thereof. Accordingly, the objection to the drawings as being prior art should be withdrawn.

Claim Objections

Claims 10 and 23 were objected to because of informalities. Amendments to claims 10 and 23 have been made above to improve their form. It is believed that the above amendments obviate the claim objections, and therefore, the Applicant respectfully requests withdrawal of the objection.

Claim Rejections - 35 U.S.C. § 112

Claim 17 was rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claim 17 has been amended above to improve its form. In light of this amendment, it is believed that the rejection under 35 USC §112 has been obviated. Therefore, the Applicant respectfully requests withdrawal of the rejection.

Claim Rejections - 35 U.S.C. § 103

Claims 1-23 were rejected under 35 U.S.C. § 103(a) as being unpatentable over “C++: Effective Object-Oriented Software Construction” by Dattatri (hereinafter “Dattatri”), in view of “Compiler Transformations for High-Performance Computing” by Bacon et al. (hereinafter “Bacon”). Applicant respectfully traverses the § 103(a) rejections, as the Examiner has failed to establish a prima facie case of obviousness. In order to establish a prima facie case, the references must show that the cited references must teach or suggest each of the elements of the claimed invention (MPEP § 706.02(j) and 2142-43). With respect to the cited references, all elements of the independent claims are neither taught nor suggested.

The pending claims relate to a method or apparatus for optimizing indirect method invocation at a call site. The call site is associated with a receiver object, and programmed to call a target method of a plurality of possible target methods that return constant values. The need for a function call at these sites is thus obviated. The invention obviates the need for the function call by generating a return constant table which has constants has table entries, the constants are those that will be returned based on specific calls. Next, during compile time, instructions that would normally call the function that generates the constant are replaced with an optimized instruction that merely retrieves the constant value. As a result of this generated “return constant table” and the “optimized instruction,” there is no need to call the function.

Dattatri relates to a system or method that makes function calls more efficient by replacing an explicit call with a pointer to a corresponding virtual table pointer. The virtual table pointer is retrieved, the index to the address of the function call is referenced, and the function’s address used within the code flow to quickly call the function. As a result, although the function is still

called, the process is generally more efficient in identifying and accessing the function code location.

Dattatri specifically does not, however, generate a return constant table. Setting up a table of pointers to functions is clearly not the same as setting up a table of constants. Moreover, Dattatri does not generate an optimized instruction to return a constant value as required by the claims. Contrary to the Examiner's assertions, the example provided from Dattatri in the most recent Office Action of "generating an optimized instruction in association with the call site to retrieve a return value associated with the target method", which referred specifically to the top of page 707 ("This (pb->g()) can be written as (*(pb->vptr[2])) () ;") does not, in fact, optimize an instruction to return a constant to thereby replace the function call as required by the claims.. In the cited line, the compiler has simply replaced the function call with a pointer to its corresponding vtbl entry. Although this substitution improves the efficiency of the function call, it does not circumvent the function call. Indeed, the function call still takes place just as before, albeit without the additional level of lookup indirection. Further, the cited function in Dattatri does not necessarily even "retrieve a value." The continued reliance on a function call is made clear later on the same page, where the comment for step B states "go to the index where the address of g() is stored," and for step C: "jump to the function whose address is stored here." Each of the independent claims of the present application, i.e., claims 1, 12, and 22 relate to the generation of the return constant table and an optimized instruction that retrieves a return value, wherein the optimized instruction is used to **replace** the function call. Given the way Dattatri uses a table to set up virtual function calls, and hence function calls, Dattatri simply does not teach or suggest the use of either the return constant table or the optimized instruction.

Bacon does not satisfy the inadequacies of Dattatri. Bacon relates to the construction of a cache to store recent invocation results. Bacon does not generate a constant return table, nor does it create an optimized instruction. Instead, it requires analysis of usage data to identify commonly-encountered function/argument pairings. In contrast, the present invention offers coverage of potentially optimizable call sites without the analysis of usage data. Indeed, Bacon specifically relies on the function calls to generate the cache of recent results where the present invention replaces the function calls. Thus, it is clear that Bacon does not teach or suggest any of the limitations of the presently pending independent claims.

The combination of Dattatri and Bacon simply does not teach or suggest each of the elements of the claimed invention. Neither Dattatri or Bacon, alone or in combination, disclose generation of return constant table or an optimized instruction in lieu of a function call as recited in independent claims 1, 12, and 22. Given that these references, either alone or in combination, fail to disclose, teach or suggest all the claim limitations, they cannot, as a matter of law, render the independent claims obvious. Reconsideration of the § 103(a) rejections is therefore respectfully requested.

Claims 2-11, 13-21, and 23 depend on these independent claims, and thus should be allowed for at least the same reasons, namely that the combination of the cited references does not teach the present invention.

Since the remarks above are believed to distinguish over the applied reference, any remaining arguments supporting the claim rejections are not acquiesced to because they are not addressed herein.

CONCLUSION

As originally filed, there were 23 claims, 3 of which were independent. As presently amended, there are 23 claims, 3 of which are independent. It is believed that no further fees are due with this Response. However, the Commissioner is hereby authorized to charge any deficiencies or credit any overpayment with respect to this patent application to deposit account number 13-2725.

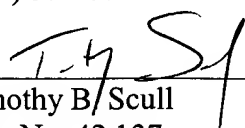
In view of the above amendments and remarks, it is believed that the application is now in condition for allowance, and such action is respectfully requested. If any additional issues need to be resolved, the Examiner is requested to telephone the undersigned to attempt to resolve those issues.

Respectfully submitted,

MERCHANT & GOULD P.C.
P.O. Box 2903
Minneapolis, Minnesota 55402-0903
(612) 332-5300

Date: June 24, 2004





Timothy B. Scull
Reg. No. 42,137